Pergamon
Computers Math. Applic. Vol. 28, No. 10–12, pp. 147–154, 1994

0898-1221/94 $7.00 + 0.00
0898-1221(94)00190-1

Method of Lines Solution of the Korteweg-de Vries Equation

W. E. Schiesser
Lehigh University, 111 Research Drive
Bethlehem, PA 18015, U.S.A.

Abstract—The Korteweg-de Vries equation (KdVE) is a classical nonlinear partial differential equation (PDE) originally formulated to model shallow water flow. In addition to the applications in hydrodynamics, the KdVE has been studied to elucidate interesting mathematical properties. In particular, the KdVE balances front sharpening and dispersion to produce solitons, i.e., traveling waves that do not change shape or speed. In this paper, we compute a solution of the KdVE by the method of lines (MOL) and compare this numerical solution with the analytical solution of the KdVE. In a second numerical solution, we demonstrate how solitons of the KdVE traveling at different velocities can merge and emerge. The numerical procedure described in the paper demonstrates the ease with which the MOL can be applied to the solution of PDEs using established numerical approximations implemented in library routines.

1. THE KdVE

The classical KdVE is [1]

$$u_t + 6uu_x + u_{xxx} = 0, \tag{1}$$

where subscripts in t and x denote partial derivatives with respect to these independent variables, e.g., $u_t = \frac{\partial u}{\partial t}$, $u_{xxx} = \frac{\partial^3 u}{\partial x^3}$; t is an initial value variable and x is a boundary value variable. Therefore, equation (1) requires one initial condition and three boundary conditions.

Equation (1) has the exact solution

$$u(x,t) = f(x-ct) = \frac{1}{2}\, c \operatorname{sech}^2 \left\{ \frac{1}{2} \sqrt{c}\,(x-ct) \right\}, \tag{2}$$

which is the equation for a soliton traveling from left to right with velocity c and height $(1/2)\,c$.

We take as the initial condition for equation (1)

$$u(x,0) = f(x) = \frac{1}{2}\, c \operatorname{sech}^2 \left\{ \frac{1}{2} \sqrt{c}\,(x) \right\}, \tag{3}$$

which follows directly from equation (2) with $t = 0$.

The boundary conditions required by equation (1) are not used in the calculation of the numerical solution. Rather, we choose an interval in x which is essentially infinite (but, of course, is finite when used in the computer code). Specifically, we use in place of the infinite interval $-\infty \le x \le \infty$ the finite interval $-30 \le x \le 70$. Since the computed solitons do not closely approach these finite boundaries, i.e., $x = -30$ and $x = 70$, the imposition of boundary conditions is not required (thus, we have the somewhat unexpected situation that a numerical solution is easier to compute for an infinite interval than for a finite interval).

2. MOL SOLUTION OF THE KdV

The essential features of a MOL solution of equations (1) and (3) are [2]:

(1) The discretization of the spatial derivatives, u_x and u_{xxx}, in equation (1).

(2) The integration of the temporal derivative, u_t, in equation (1), which requires the integration of a system of ordinary differential equations (ODEs) in t as a result of the spatial discretization of feature (1) above.

We focus attention on feature (1), and accomplish feature (2) with an established ODE integrator, RKF45 [3].

The derivative u_x is computed by finite differences in two ways [2]:

(1) five point biased upwind approximations implemented in library subroutine DSS020, and

(2) five point centered approximations implemented in library subroutine DSS004.

The derivative u_{xxx} is computed from a seven point centered approximation reported by Fornberg [4], which is implemented in subroutine UXXX7C.

A subroutine for calculating the MOL ODE temporal derivatives is listed in Program 1:

```
      SUBROUTINE DERV
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      PARAMETER (NG=400)
      COMMON/T/          T,      NSTOP,      NORUN
     1      /Y/   U(0:NG)
     2      /F/  UT(0:NG)
     3      /S/  UX(0:NG),UXXX(0:NG),    X(0:NG)
     4      /C/         XL,        XR,          C,        SRC
     5      /I/         IP
C...
C...  U
C...   X
         IF(NORUN.EQ.1)CALL DSS020(XL,XR,NG+1,U,UX,1.0D0)
         IF(NORUN.EQ.2)CALL DSS004(XL,XR,NG+1,U,UX)
C...
C...  U
C...   XXX
         CALL UXXX7C(XL,XR,NG+1,U,UXXX)
C...
C...  ODES
      DO 10 I=0,NG
C...
C...     PDE
         UT(I)=-6.0D0*U(I)*UX(I)-UXXX(I)
10    CONTINUE
      RETURN
      END
```

Program 1. Subroutine DERV for the calculation of the temporal derivatives of equation (1).

The following points can be noted about subroutine DERV (reading from top to bottom):

1. The number of spatial intervals in x, NG, is 400.
2. u is in array U(0:NG) in COMMON/Y/.
3. u_t is in array UT(0:NG) in COMMON/F/.

4. u_x and u_{xxx} are in arrays UX(0:NG) and UXXX(0:NG), respectively, in COMMON/S/.
5. u_x is computed by a call to DSS020 for the first solution (NORUN = 1) or a call to DSS004 for the second solution (NORUN = 1).

```
C...
C...   U
C...    X
          IF(NORUN.EQ.1)CALL DSS020(XL,XR,NG+1,U,UX,1.0D0)
          IF(NORUN.EQ.2)CALL DSS004(XL,XR,NG+1,U,UX)
```

   The details of these routines are given elsewhere [2].
6. u_{xxx} is then computed by a call to UXXX7C

```
C...
C...   U
C...    XXX
          CALL UXXX7C(XL,XR,NG+1,U,UXXX)
```

   Subroutine UXXX7C is listed in the Appendix to illustrate: (a) the use of a tabulated finite difference in a MOL differentiation routine, in this case, a seven point finite difference for u_{xxx} given by Fornberg [4], and (b) the programming for an infinite interval in x which does not require the imposition of specific boundary conditions.
7. u_t is calculated in DO loop 10

```
C...
C...   ODES
       DO 10 I=0,NG
C...
C...      PDE
          UT(I)=-6.0D0*U(I)*UX(I)-UXXX(I)
10     CONTINUE
```

   This coding illustrates one of the positive features of the MOL, the close resemblance between the PDE(s), in this case equation (1), and the coding. Also, the coding is very compact considering the complexity and nonlinearity of the PDE.
8. The 401 ODE temporal derivatives are sent to the ODE integrator, RKF45, through COMMON/F/, and the 401 dependent variables are returned to DERV through COMMON/Y/ for use in the programming of the MOL approximation of the PDE.

The numerical and analytical solutions are printed and plotted in an output routine. The plotted output is given in Figure 1 for $c = 1$ (unit velocity) and NORUN = 1, which shows the solitons traveling left to right for $t = 0$ (centered at $x = 0$), $5, 10, \ldots 35$ (centered at $x = 35$). This comparison of the solutions is particularly interesting at the peak of the solitons (for $x = ct$ in equation (2)), which, from equation (2), has the value 1/2. This comparison is given in Table 1, which lists the numerical and analytical solutions at $x - ct = -0.25, 0, 0.25$ for $t = 0$ and 30:

Table 1. Comparison of the numerical and analytical solutions to equation (1) near $x = ct$.

DSS020 (NORUN = 1)

I	T	X(I)	X(I)-T	ABS(UN)	ABS(UE)	DIFF
119	0.00	-0.25	-0.250	0.49227	0.49227	0.000D+00
120	0.00	0.00	0.000	0.50000	0.50000	0.000D+00
121	0.00	0.25	0.250	0.49227	0.49227	0.000D+00

```
CONSERVATION OF MASS   =  2.0000
CONSERVATION OF ENERGY =  0.3333
WHITHAM CONSERVATION   =  0.4000
```

I	T	X(I)	X(I)-T	ABS(UN)	ABS(UE)	DIFF
239	30.00	29.75	-0.250	0.49245	0.49227	0.186D-03
240	30.00	30.00	0.000	0.49923	0.50000	-0.765D-03
241	30.00	30.25	0.250	0.49054	0.49227	-0.172D-02

```
CONSERVATION OF MASS   =  1.9990
CONSERVATION OF ENERGY =  0.3323
WHITHAM CONSERVATION   =  0.3979
```

DSS004 (NORUN = 2)

I	T	X(I)	X(I)-T	ABS(UN)	ABS(UE)	DIFF
119	0.00	-0.25	-0.250	0.49227	0.49227	0.000D+00
120	0.00	0.00	0.000	0.50000	0.50000	0.000D+00
121	0.00	0.25	0.250	0.49227	0.49227	0.000D+00

```
CONSERVATION OF MASS   =  2.0000
CONSERVATION OF ENERGY =  0.3333
WHITHAM CONSERVATION   =  0.4000
```

I	T	X(I)	X(I)-T	ABS(UN)	ABS(UE)	DIFF
239	30.00	29.75	-0.250	0.49230	0.49227	0.325D-04
240	30.00	30.00	0.000	0.50020	0.50000	0.196D-03
241	30.00	30.25	0.250	0.49212	0.49227	-0.148D-03

```
CONSERVATION OF MASS   =  2.0000
CONSERVATION OF ENERGY =  0.3333
WHITHAM CONSERVATION   =  0.3999
```

We can note the following points about the output in Table 1:

1. At $t = 0$, the solution is 0.50000 at $x = 0$, as expected (in accordance with equation (2)). This peak value is then maintained by the soliton when $x = ct$ in accordance with equation (2), e.g., for $c = 1$, $x = t = 30$, the analytical solution is again 0.50000.
2. The solutions of equation (1) satisfy an infinity of conservation principles. Here, we illustrate the calculation of three:

   (a) Conservation of mass, defined as:

   $$u_1(t) = \int_{-\infty}^{\infty} u(x,t)\,dx. \tag{4}$$

   (b) Conservation of energy, defined as:

   $$u_2(t) = \int_{-\infty}^{\infty} \frac{1}{2}\,u^2(x,t)\,dx. \tag{5}$$

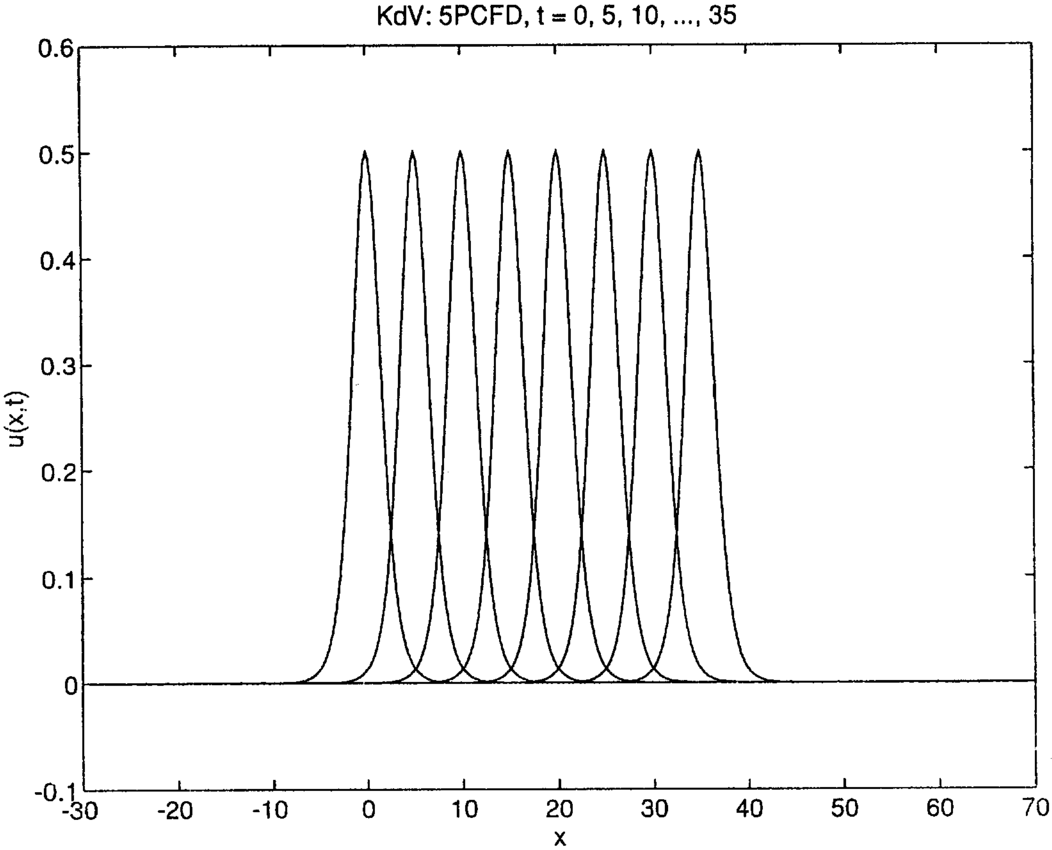

Figure 1. MOL solution to equation (1) with initial condition equation (3).

(c) Conservation proposed by Whitham [1], defined as:

$$u_3(t) = \int_{-\infty}^{\infty} 2u^3(x,t) - u_x^2(x,t)\, dx. \tag{6}$$

Integrals $u_1(t)$, $u_2(t)$, and $u_3(t)$ were evaluated numerically by Simpson's rule. From the numerical output, we see $u_1(0) = 2$, $u_2(0) = 1/3$, and $u_3(0) = 0.4$.

In using the five point biased upwind approximations (5PBUFD) in DSS020, errors accumulated with increasing t so that the three integrals of equations (4), (5), and (6) were correct to about three figures at $t = 30$, i.e., $u_1(30) = 1.990$, $u_2(30) = 0.3323$, and $u_3(30) = 0.3979$.

In using the five point centered approximations (5PCFD) in DSS004, errors accumulated with increasing t so that the three integrals of equations (4), (5), and (6) were correct to four figures at $t = 30$, i.e., $u_1(30) = 2.000$, $u_2(30) = 0.3333$, and $u_3(30) = 0.3999$.

Thus, the centered approximations in this case performed better than the biased upwind approximations, even though the solution of equation (1) appears to be "strongly convective" as suggested by Figure 1, and therefore one would expect that some upwinding would lead to better results.

3. The better performance of the centered approximations is also evident in the numerical solutions. For the biased upwind approximations, the differences between the numerical solution, $u_n(x,t)$, and the analytical solution, $u_e(x,t)$, that is $\Delta(x,t) = u_n(x,t) - u_e(x,t)$, are

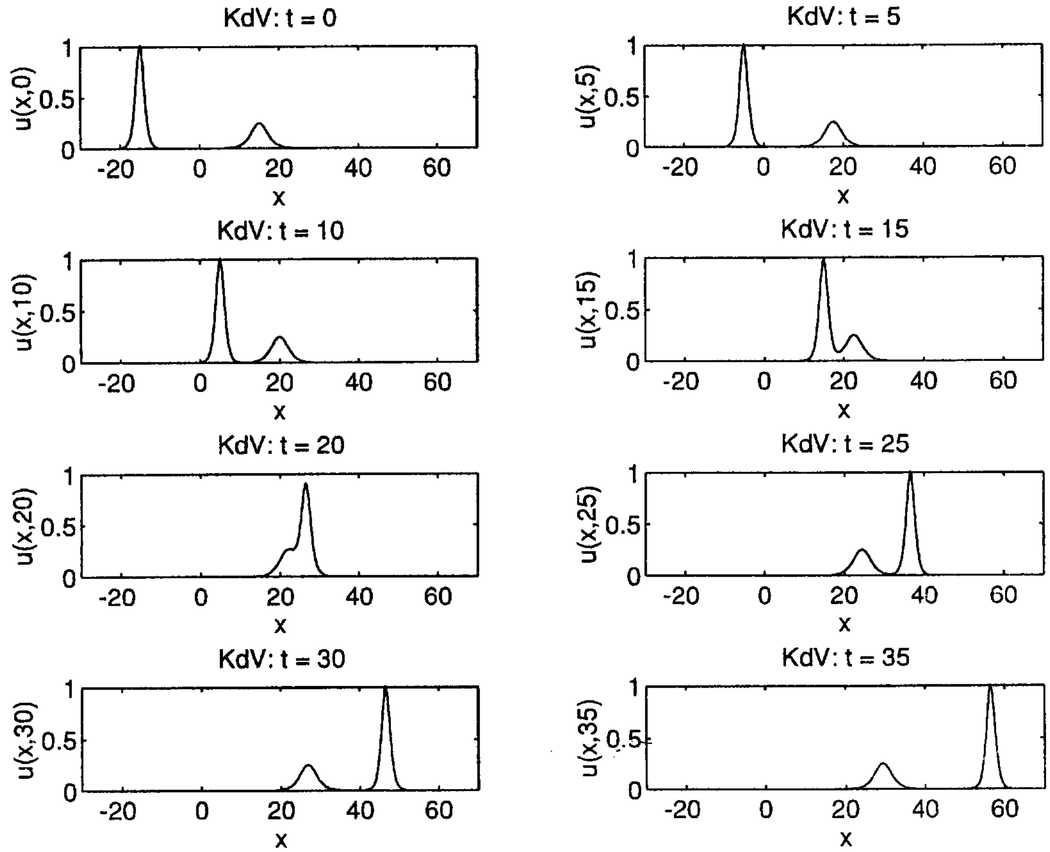

Figure 2. MOL solution to equation (1) with two initial pulses (from equation (3)).

Biased upwind (DSS020)
$\Delta(29.75, 30) = 0.000186$
$\Delta(30.00, 30) = -0.000765$
$\Delta(30.25, 30) = -0.00172$

Centered (DSS004)
$\Delta(29.75, 30) = 0.0000325$
$\Delta(30.00, 30) = 0.000196$
$\Delta(30.25, 30) = -0.000148$

The centered approximations gave substantially smaller errors than the biased upwind approximations.

A second MOL solution was computed for an initial condition consisting of the sum of two "sech" pulses, i.e., pulses of the form given by equation (3): (a) a pulse centered at $x = -15$ with $c = 2$, and (b) a pulse centered at $x = 15$ with $c = 0.5$, as plotted in Figure 2 ($t = 0$). The left pulse has a higher velocity (and therefore also height) so that it overtakes and merges with the right pulse, as indicated in Figure 2. The faster pulse then emerges to the right of the slower pulse. Eventually, the two original pulses (at $t = 0$) reappear, e.g., at $t = 35$ as indicated in Figure 2, and continue to travel with their original shape.

3. SUMMARY

The MOL has been used to compute a solution to the KdVE with modest programming. Different approximations could easily be used by switching between library routines (i.e., DSS020 vs. DSS004). The temporal integration of the 401 ODEs was easily accomplished with a library explicit integrator, RKF45 (although the use of an implicit integrator might generally be required if the ODEs are stiff). Based on this experience and that of many previous studies, we can recommend the MOL as a convenient method for the numerical integration of PDEs.

A complete, documented Fortran code for the solution of the KdVE, including all of the library routines discussed in this paper, is available on request from the author on a DOS-formatted 1.4 mb, 3.5 inch diskette. This code can be used to investigate variants of the KdVE, e.g.,

$$u_t + 6u^n u_x + u_{xxx} = 0, \tag{7}$$

which undergoes substantial changes in the solution for $n \geq 4$ [5]. Also, the code can be modified for the solution of other PDEs.

APPENDIX
SUBROUTINE UXXX7C

Subroutine UXXX7C is listed below to illustrate how existing approximations for spatial derivatives can be used within the MOL.

```
      SUBROUTINE UXXX7C(XL,XU,N,U,UXXX)
C...
C...  DOUBLE PRECISION CODING IS USED
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C...
C...  VARIABLE DIMENSION ARRAYS
      DIMENSION U(N), UXXX(N)
C...
C...  GRID SPACING, 1/(8*(DX**3))
      DX=(XU-XL)/DFLOAT(N-1)
      R8DX=1.0D0/(8.0D0*(DX**3))
C...
C...  COMPUTE THIRD SPATIAL DERIVATIVE
      DO 1 I=1,N
C...
C...     AT THE LEFT END, UXXX = 0
         IF(I.LT.4)THEN
         UXXX(I)=0.0D0
C...
C...     AT THE RIGHT END, UXXX = 0
         ELSE
     +   IF(I.GT.(N-3))THEN
         UXXX(I)=0.0D0
C...
C...     INTERIOR POINTS (FIVE POINT CENTERED APPROXIMATION; SEE FORNBERG,
C...     B, "FAST GENERATION OF WEIGHTS IN FINITE DIFFERENCE FORMULAS", IN
C...     RECENT DEVELOPMENTS IN NUMERICAL METHODS AND SOFTWARE FOR ODES/
C...     DAES/PDES, G. D. GYRNE AND W. E. SCHIESSER (EDS.), WORLD SCIENT-
C...     IFIC, RIVER EDGE, NJ, 1992, FIG. 3, P114.
         ELSE
         UXXX(I)=R8DX*
     1   (  1.D+00      *U(I-3)
     2     -8.D+00      *U(I-2)
```

```
     3       +13.D+00       *U(I-1)
     4        +0.D+00       *U(I )
     5       -13.D+00       *U(I+1)
     6        +8.D+00       *U(I+2)
     6        -1.D+00       *U(I+3))
           END IF
C...
C...    NEXT GRID POINT
1       CONTINUE
        RETURN
        END
```

REFERENCES

1. G. Strang, *Introduction to Applied Mathematics*, pp. 599–602, Wellesley-Cambridge Press, Wellesley, MA, (1986).
2. W.E. Schiesser, *The Numerical Method of Lines Integration of Partial Differential Equations*, Academic Press, San Diego, (1991).
3. G.E. Forsythe, M.A. Malcolm and C.B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Englewood Cliffs, NJ, (1977).
4. B. Fornberg, Fast generation of weights in finite difference formulas, In *Recent Developments in Numerical Methods and Software for ODEs/DAEs/PDEs*, (Edited by G.D. Byrne and W.E. Schiesser), p. 114, Figure 3, World Scientific, River Edge, NJ, (1992).
5. T. Taha and J. Liaw, A parallel algorithm for an investigation of a self-focusing singularity of higher KdV equations, In *Fifth International Symposium on Domain Decomposition Methods for PDES*, (Edited by D. Keyes *et al.*), p. 597–604, SIAM, Philadelphia, PA, (1992).